

# CS:4350 Logic in Computer Science

## Propositional Logic

Cesare Tinelli

Spring 2021



# Credits

These slides are largely based on slides originally developed by **Andrei Voronkov** at the University of Manchester. Adapted by permission.

# Propositional Logic

- **Syntax:** set of formulas built with propositional variables and connectives
- **Semantics:** formulas are assigned a Boolean value (true, false)
- **Inference system:** several

The sentences of the language (formulas) are also called *propositions*

# Propositional Logic

- **Syntax:** set of formulas built with propositional variables and connectives
- **Semantics:** formulas are assigned a Boolean value (true, false)
- **Inference system:** several

The sentences of the language (formulas) are also called *propositions*

# Propositions in mathematical logic

Formalize **natural language statements** that can be **either true or false**

Basic propositions are called *atomic*

Examples:

1.  $0 < 1$
2. Alan Turing was born in Manchester
3.  $1 + 1 = 10$

More **complex propositions** are built from simpler ones via small number of constructs

Examples:

1. If  $0 < 1$  then Alan Turing was born in Manchester
2.  $1 + 1 = 10$  or  $1 + 1 \neq 10$

# Propositions in mathematical logic

Formalize **natural language statements** that can be **either true or false**

Basic propositions are called *atomic*

**Examples:**

1.  $0 < 1$
2. Alan Turing was born in Manchester
3.  $1 + 1 = 10$

More **complex propositions** are built from simpler ones via small number of constructs

**Examples:**

1. If  $0 < 1$  then Alan Turing was born in Manchester
2.  $1 + 1 = 10$  or  $1 + 1 \neq 10$

# Propositions in mathematical logic

Formalize **natural language statements** that can be **either true or false**

Basic propositions are called *atomic*

Examples:

1.  $0 < 1$
2. Alan Turing was born in Manchester
3.  $1 + 1 = 10$

More **complex propositions** are built from simpler ones via small number of constructs

Examples:

1. **If**  $0 < 1$  **then** Alan Turing was born in Manchester
2.  $1 + 1 = 10$  **or**  $1 + 1 \neq 10$

# Propositions in mathematical logic

Formalize **natural language statements** that can be **either true or false**

Basic propositions are called *atomic*

**Examples:**

1.  $0 < 1$
2. Alan Turing was born in Manchester
3.  $1 + 1 = 10$

More **complex propositions** are built from simpler ones via small number of constructs

**Examples:**

1. **If**  $0 < 1$  **then** Alan Turing was born in Manchester
2.  $1 + 1 = 10$  **or**  $1 + 1 \neq 10$



# Truth of atomic propositions

Each proposition formalizes a statement that is either **true** or **false**

The *truth value* (true, false) of an atomic proposition  $P$  depends on  $P$ 's *interpretation*

**Example** What is the true value of  $1 + 1 = 10$ ?

- false, if we interpret 1 and 10 as integers in decimal notation, and + as addition
- true, if we interpret 1 and 10 as integers in binary notation, and + as addition

# Truth of atomic propositions

Each proposition formalizes a statement that is either true or false

The *truth value* (true, false) of an atomic proposition  $P$  depends on  $P$ 's *interpretation*

Example What is the true value of  $1 + 1 = 10$ ?

- false, if we interpret 1 and 10 as integers in decimal notation, and + as addition
- true, if we interpret 1 and 10 as integers in binary notation, and + as addition

# Truth of atomic propositions

Each proposition formalizes a statement that is either true or false

The *truth value* (true, false) of an atomic proposition  $P$  depends on  $P$ 's *interpretation*

**Example** What is the true value of  $1 + 1 = 10$ ?

- false, if we interpret 1 and 10 as integers in decimal notation, and + as addition
- true, if we interpret 1 and 10 as integers in binary notation, and + as addition

# Truth of atomic propositions

Each proposition formalizes a statement that is either true or false

The *truth value* (true, false) of an atomic proposition  $P$  depends on  $P$ 's *interpretation*

**Example** What is the true value of  $1 + 1 = 10$ ?

- **false**, if we interpret 1 and 10 as integers in decimal notation, and + as addition
- true, if we interpret 1 and 10 as integers in binary notation, and + as addition

# Truth of atomic propositions

Each proposition formalizes a statement that is either true or false

The *truth value* (true, false) of an atomic proposition  $P$  depends on  $P$ 's *interpretation*

**Example** What is the true value of  $1 + 1 = 10$ ?

- **false**, if we interpret 1 and 10 as integers in decimal notation, and + as addition
- **true**, if we interpret 1 and 10 as integers in binary notation, and + as addition

## Truth of complex propositions

Consider a **complex proposition**  $C$  built with a construct  $c$  from simpler propositions  $S_1, \dots, S_n$

The truth value of  $C$  depends

1. the meaning of  $c$
2. the truth value of  $S_1, \dots, S_n$

More precisely, it is a function (namely, the meaning of  $c$ ) of truth values of  $S_1, \dots, S_n$

**Example**  $(1 + 1 = 10 \text{ or } 1 + 1 \neq 10)$  is true if at least one of  $1 + 1 = 10$ ,  $1 + 1 \neq 10$  is true

## Truth of complex propositions

Consider a **complex proposition**  $C$  built with a construct  $c$  from simpler propositions  $S_1, \dots, S_n$

The truth value of  $C$  depends

1. the meaning of  $c$
2. the truth value of  $S_1, \dots, S_n$

More precisely, it is a **function** (namely, the meaning of  $c$ ) of truth values of  $S_1, \dots, S_n$

**Example**  $(1 + 1 = 10 \text{ or } 1 + 1 \neq 10)$  is true if at least one of  $1 + 1 = 10$ ,  $1 + 1 \neq 10$  is true

## Truth of complex propositions

Consider a **complex proposition**  $C$  built with a construct  $c$  from simpler propositions  $S_1, \dots, S_n$

The truth value of  $C$  depends

1. the meaning of  $c$
2. the truth value of  $S_1, \dots, S_n$

More precisely, it is a **function** (namely, the meaning of  $c$ ) of truth values of  $S_1, \dots, S_n$

Example ( $1 + 1 = 10$  or  $1 + 1 \neq 10$ ) is true if at least one of  $1 + 1 = 10$ ,  $1 + 1 \neq 10$  is true



## Truth of complex propositions

Consider a **complex proposition**  $C$  built with a construct  $c$  from simpler propositions  $S_1, \dots, S_n$

The truth value of  $C$  depends

1. the meaning of  $c$
2. the truth value of  $S_1, \dots, S_n$

More precisely, it is a **function** (namely, the meaning of  $c$ ) of truth values of  $S_1, \dots, S_n$

**Example**  $(1 + 1 = 10 \text{ or } 1 + 1 \neq 10)$  is true if at least one of  $1 + 1 = 10$ ,  $1 + 1 \neq 10$  is true

# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

## *Propositional formula:*

- Every propositional variable (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula

# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

*Propositional formula:*

- Every propositional **variable** (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula

# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

*Propositional formula:*

- Every propositional variable (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula

# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

*Propositional formula:*

- Every propositional variable (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula

# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

*Propositional formula:*

- Every propositional variable (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula

# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

*Propositional formula:*

- Every propositional variable (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula

# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

*Propositional formula:*

- Every propositional variable (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula



# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

*Propositional formula:*

- Every propositional *variable* (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula

The symbols  $\top, \perp, \wedge, \vee, \neg, \rightarrow, \leftrightarrow$  are called *connectives*

# Propositional logic: syntax

Assume a countable set of *propositional variables* ( $\{p, p_1, p_2, \dots, q, q_1, q_2, \dots\}$ )

*Propositional formula:*

- Every propositional variable (aka, *atom*) is a formula
- $\top$  and  $\perp$  are formulas
- If  $A_1, \dots, A_n$  are formulas, where  $n \geq 2$ , then  $A_1 \wedge \dots \wedge A_n$  and  $A_1 \vee \dots \vee A_n$  are formulas
- If  $A$  is a formula, then  $\neg A$  is a formula
- If  $A$  and  $B$  are formulas, then  $A \rightarrow B$  and  $A \leftrightarrow B$  are formulas
- Nothing else is a formula

The symbols  $\top, \perp, \wedge, \vee, \neg, \rightarrow, \leftrightarrow$  are called *connectives*

**Note:** The book considers also  $\oplus$  (exclusive or),  $\downarrow$  (nor), and  $\uparrow$  (nand)

# Parsing expressions

In general, we use **parentheses** to disambiguate the parsing of expressions

Parenthesis clutter can be reduced by assigning precedence to operators

**Example** In arithmetic we know that the expression

$$x \cdot y + 2 \cdot z \quad \text{stands for} \quad (x \cdot y) + (2 \cdot z)$$

since  $\cdot$  has a higher precedence than  $+$

## Parsing expressions

In general, we use **parentheses** to disambiguate the parsing of expressions

Parenthesis clutter can be reduced by assigning precedence to operators

**Example** In arithmetic we know that the expression

$$x \cdot y + 2 \cdot z \quad \text{stands for} \quad (x \cdot y) + (2 \cdot z)$$

since  $\cdot$  has a higher precedence than  $+$

## Parsing expressions

In general, we use **parentheses** to disambiguate the parsing of expressions

Parenthesis clutter can be reduced by assigning precedence to operators

**Example** In arithmetic we know that the expression

$$x \cdot y + 2 \cdot z \quad \text{stands for} \quad (x \cdot y) + (2 \cdot z)$$

since  $\cdot$  has a **higher precedence** than  $+$

# Propositional connectives and their precedence

Connective	Name	Precedence
$\top$	verum	
$\perp$	falsum	
$\neg$	negation	5
$\wedge$	conjunction	4
$\vee$	disjunction	3
$\rightarrow$	implication	2
$\leftrightarrow$	equivalence	1

Implication is right-associative:

$A \rightarrow B \rightarrow C$  is parsed as

$A \rightarrow (B \rightarrow C)$

# Propositional connectives and their precedence

Connective	Name	Precedence
$\top$	verum	
$\perp$	falsum	
$\neg$	negation	5
$\wedge$	conjunction	4
$\vee$	disjunction	3
$\rightarrow$	implication	2
$\leftrightarrow$	equivalence	1

Implication is **right-associative**:

$A \rightarrow B \rightarrow C$  is parsed as

$A \rightarrow (B \rightarrow C)$

# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\neg$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Inside-out (starting with the highest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Outside-in (starting with the lowest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$



# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\top$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Inside-out (starting with the highest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Outside-in (starting with the lowest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\top$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

**Inside-out** (starting with the **highest precedence** connectives):

$$(((\neg A) \wedge B) \rightarrow (C \vee D)) \leftrightarrow E$$

**Outside-in** (starting with the lowest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\top$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Inside-out (starting with the highest precedence connectives):

$$((\neg A) \wedge B) \rightarrow (C \vee D) \leftrightarrow E$$

Outside-in (starting with the lowest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\top$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Inside-out (starting with the highest precedence connectives):

$$(((\neg A) \wedge B) \rightarrow (C \vee D)) \leftrightarrow E$$

Outside-in (starting with the lowest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\top$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Inside-out (starting with the highest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Outside-in (starting with the **lowest precedence** connectives):

$$(((\neg A) \wedge B) \rightarrow (C \vee D)) \leftrightarrow E$$

# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Inside-out (starting with the highest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Outside-in (starting with the lowest precedence connectives):

$$(((\neg A) \wedge B) \rightarrow (C \vee D)) \leftrightarrow E$$

# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\top$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Inside-out (starting with the highest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Outside-in (starting with the lowest precedence connectives):

$$(((\neg A) \wedge B) \rightarrow (C \vee D)) \leftrightarrow E$$

# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\top$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Inside-out (starting with the highest precedence connectives):

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Outside-in (starting with the lowest precedence connectives):

$$(((\neg A) \wedge B) \rightarrow (C \vee D)) \leftrightarrow E$$



# Parsing Formulas

Let us parse

$$\neg A \wedge B \rightarrow C \vee D \leftrightarrow E$$

Connective	Precedence
$\neg$	
$\perp$	
$\neg$	5
$\wedge$	4
$\vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

**Inside-out** (starting with the **highest precedence** connectives):

$$(((\neg A) \wedge B) \rightarrow (C \vee D)) \leftrightarrow E$$

**Outside-in** (starting with the **lowest precedence** connectives):

$$(((\neg A) \wedge B) \rightarrow (C \vee D)) \leftrightarrow E$$

# Semantics, Interpretation

In arithmetic the meaning of expressions with variables like

$$x \cdot y + 2 \cdot z$$

is defined as follows:

1. fix a mapping from variables to (integer) values,  
e.g.,  $\{x \mapsto 1, y \mapsto 7, z \mapsto -3\}$
2. then, under this mapping the expression has the value 1

# Semantics, Interpretation

In arithmetic the meaning of expressions with variables like

$$x \cdot y + 2 \cdot z$$

is defined as follows:

1. fix a **mapping from variables to (integer) values**,

e.g.,  $\{ x \mapsto 1, y \mapsto 7, z \mapsto -3 \}$

2. then, under this mapping the expression has the value 1

# Semantics, Interpretation

In arithmetic the meaning of expressions with variables like

$$x \cdot y + 2 \cdot z$$

is defined as follows:

1. fix a **mapping from variables to (integer) values**,  
e.g.,  $\{x \mapsto 1, y \mapsto 7, z \mapsto -3\}$
2. then, under this mapping the expression has the value 1

# Semantics, Interpretation

In arithmetic the meaning of expressions with variables like

$$x \cdot y + 2 \cdot z$$

is defined

**In other words:**

1. fix a mapping from variables to values  
e.g.,
2. then, under this mapping the expression has the value 1

# Semantics, Interpretation

The semantics of propositional formulas can be defined too by **assigning values to variables**

- There are two *Boolean/truth values*:  
true (denoted by 1) and false (denoted by 0)
- An *interpretation* for a set  $P$  of propositional variables is a mapping  $\mathcal{I} : P \rightarrow \{1, 0\}$
- Interpretations are also called *truth assignments*

# Semantics, Interpretation

The semantics of propositional formulas can be defined too by **assigning values to variables**

- There are two *Boolean/truth values*:  
**true** (denoted by **1**) and **false** (denoted by **0**)
- An *interpretation* for a set  $P$  of propositional variables is a mapping  $\mathcal{I} : P \rightarrow \{1, 0\}$
- Interpretations are also called *truth assignments*

# Semantics, Interpretation

The semantics of propositional formulas can be defined too by **assigning values to variables**

- There are two *Boolean/truth values*:  
**true** (denoted by **1**) and **false** (denoted by **0**)
- An *interpretation* for a set  $P$  of propositional variables is a mapping  $\mathcal{I} : P \rightarrow \{1, 0\}$
- Interpretations are also called *truth assignments*



# Semantics, Interpretation

The semantics of propositional formulas can be defined too by *assigning values to variables*

- There are two *Boolean/truth values*:  
*true* (denoted by 1) and *false* (denoted by 0)
- An *interpretation* for a set  $P$  of propositional variables is a mapping  $\mathcal{I} : P \rightarrow \{1, 0\}$
- Interpretations are also called *truth assignments*

## Interpreting formulas

The truth value of a **complex formula** is determined by the truth values of its components

An interpretation  $\mathcal{I}$  extends to a mapping from all formulas to truth values as follows:

# Interpreting formulas

The truth value of a complex formula is determined by the truth values of its components

An interpretation  $\mathcal{I}$  extends to a mapping from all formulas to truth values as follows:

1.  $\mathcal{I}(\top) = 1$  and  $\mathcal{I}(\perp) = 0$
2.  $\mathcal{I}(A_1 \wedge \dots \wedge A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for all  $i$
3.  $\mathcal{I}(A_1 \vee \dots \vee A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for some  $i$
4.  $\mathcal{I}(\neg A) = 1$  iff  $\mathcal{I}(A) = 0$
5.  $\mathcal{I}(A_1 \rightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = 0$  or  $\mathcal{I}(A_2) = 1$
6.  $\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

# Interpreting formulas

The truth value of a **complex formula** is determined by the truth values of its components

An interpretation  $\mathcal{I}$  **extends** to a mapping from **all** formulas to truth values as follows:

1.  $\mathcal{I}(\top) = 1$  and  $\mathcal{I}(\perp) = 0$
2.  $\mathcal{I}(A_1 \wedge \dots \wedge A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for all  $i$
3.  $\mathcal{I}(A_1 \vee \dots \vee A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for some  $i$
4.  $\mathcal{I}(\neg A) = 1$  iff  $\mathcal{I}(A) = 0$
5.  $\mathcal{I}(A_1 \rightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = 0$  or  $\mathcal{I}(A_2) = 1$
6.  $\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

# Interpreting formulas

The truth value of a **complex formula** is determined by the truth values of its components

An interpretation  $\mathcal{I}$  **extends** to a mapping from **all** formulas to truth values as follows:

1.  $\mathcal{I}(\top) = 1$  and  $\mathcal{I}(\perp) = 0$
2.  $\mathcal{I}(A_1 \wedge \dots \wedge A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for all  $i$
3.  $\mathcal{I}(A_1 \vee \dots \vee A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for some  $i$
4.  $\mathcal{I}(\neg A) = 1$  iff  $\mathcal{I}(A) = 0$
5.  $\mathcal{I}(A_1 \rightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = 0$  or  $\mathcal{I}(A_2) = 1$
6.  $\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

# Interpreting formulas

The truth value of a **complex formula** is determined by the truth values of its components

An interpretation  $\mathcal{I}$  **extends** to a mapping from **all** formulas to truth values as follows:

1.  $\mathcal{I}(\top) = 1$  and  $\mathcal{I}(\perp) = 0$
2.  $\mathcal{I}(A_1 \wedge \dots \wedge A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for all  $i$
3.  $\mathcal{I}(A_1 \vee \dots \vee A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for some  $i$
4.  $\mathcal{I}(\neg A) = 1$  iff  $\mathcal{I}(A) = 0$
5.  $\mathcal{I}(A_1 \rightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = 0$  or  $\mathcal{I}(A_2) = 1$
6.  $\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

# Interpreting formulas

The truth value of a complex formula is determined by the truth values of its components

An interpretation  $\mathcal{I}$  extends to a mapping from all formulas to truth values as follows:

1.  $\mathcal{I}(\top) = 1$  and  $\mathcal{I}(\perp) = 0$
2.  $\mathcal{I}(A_1 \wedge \dots \wedge A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for all  $i$
3.  $\mathcal{I}(A_1 \vee \dots \vee A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for some  $i$
4.  $\mathcal{I}(\neg A) = 1$  iff  $\mathcal{I}(A) = 0$
5.  $\mathcal{I}(A_1 \rightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = 0$  or  $\mathcal{I}(A_2) = 1$
6.  $\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

# Interpreting formulas

The truth value of a **complex formula** is determined by the truth values of its components

An interpretation  $\mathcal{I}$  **extends** to a mapping from **all** formulas to truth values as follows:

1.  $\mathcal{I}(\top) = 1$  and  $\mathcal{I}(\perp) = 0$
2.  $\mathcal{I}(A_1 \wedge \dots \wedge A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for all  $i$
3.  $\mathcal{I}(A_1 \vee \dots \vee A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for some  $i$
4.  $\mathcal{I}(\neg A) = 1$  iff  $\mathcal{I}(A) = 0$
5.  $\mathcal{I}(A_1 \rightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = 0$  or  $\mathcal{I}(A_2) = 1$
6.  $\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$



# Interpreting formulas

The truth value of a **complex formula** is determined by the truth values of its components

An interpretation  $\mathcal{I}$  **extends** to a mapping from **all** formulas to truth values as follows:

1.  $\mathcal{I}(\top) = 1$  and  $\mathcal{I}(\perp) = 0$
2.  $\mathcal{I}(A_1 \wedge \dots \wedge A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for all  $i$
3.  $\mathcal{I}(A_1 \vee \dots \vee A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for some  $i$
4.  $\mathcal{I}(\neg A) = 1$  iff  $\mathcal{I}(A) = 0$
5.  $\mathcal{I}(A_1 \rightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = 0$  or  $\mathcal{I}(A_2) = 1$
6.  $\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

# Interpreting formulas

The truth value of a complex formula is determined by the truth values of its components

An interpretation  $\mathcal{I}$  extends to a mapping from all formulas to truth values as follows:

1.  $\mathcal{I}(\top) = 1$  and  $\mathcal{I}(\perp) = 0$
2.  $\mathcal{I}(A_1 \wedge \dots \wedge A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for all  $i$
3.  $\mathcal{I}(A_1 \vee \dots \vee A_n) = 1$  iff  $\mathcal{I}(A_i) = 1$  for some  $i$
4.  $\mathcal{I}(\neg A) = 1$  iff  $\mathcal{I}(A) = 0$
5.  $\mathcal{I}(A_1 \rightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = 0$  or  $\mathcal{I}(A_2) = 1$
6.  $\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

## Operation tables

$\mathcal{I}(A_1 \vee A_2) = 1$  iff  $\mathcal{I}(A_1) = 1$  or  $\mathcal{I}(A_2) = 1$

$\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

$\wedge$	1	0	$\vee$	1	0	$\neg$	
1	1	0	1	1	1	1	0
0	0	0	0	1	0	0	1
$\rightarrow$	1	0	$\leftrightarrow$	1	0		
1	1	0	1	1	0		
0	1	1	0	0	1		

Therefore, every connective can be considered as a function on truth values

## Operation tables

$\mathcal{I}(A_1 \vee A_2) = 1$  iff  $\mathcal{I}(A_1) = 1$  or  $\mathcal{I}(A_2) = 1$

$\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

$\wedge$	1	0	$\vee$	1	0	$\neg$	
1	1	0	1	1	1	1	0
0	0	0	0	1	0	0	1
$\rightarrow$	1	0	$\leftrightarrow$	1	0		
1	1	0	1	1	0		
0	1	1	0	0	1		

Therefore, every connective can be considered as a function on truth values

## Operation tables

$\mathcal{I}(A_1 \vee A_2) = 1$  iff  $\mathcal{I}(A_1) = 1$  or  $\mathcal{I}(A_2) = 1$

$\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

$\wedge$		1	0	$\vee$		1	0	$\neg$		
1		1	0	1		1	1	1		0
0		0	0	0		1	0	0		1
$\rightarrow$		1	0	$\leftrightarrow$		1	0			
1		1	0	1		1	0			
0		1	1	0		0	1			

Therefore, every connective can be considered as a function on truth values

## Operation tables

$\mathcal{I}(A_1 \vee A_2) = 1$  iff  $\mathcal{I}(A_1) = 1$  or  $\mathcal{I}(A_2) = 1$

$\mathcal{I}(A_1 \leftrightarrow A_2) = 1$  iff  $\mathcal{I}(A_1) = \mathcal{I}(A_2)$

$\wedge$		1	0	$\vee$		1	0	$\neg$		
1		1	0	1		1	1	1		0
0		0	0	0		1	0	0		1
$\rightarrow$		1	0	$\leftrightarrow$		1	0			
1		1	0	1		1	0			
0		1	1	0		0	1			

Therefore, every connective can be considered as a **function** on truth values

# Satisfiability, validity, equivalence

Let  $A$  and  $B$  be two formulas and  $\mathcal{I}$  an interpretation

- If  $\mathcal{I}(A) = 1$ , we write  $\mathcal{I} \models A$  and say, equivalently, that  $A$  is *true* in  $\mathcal{I}$ ,  $\mathcal{I}$  *satisfies*  $A$ , or  $\mathcal{I}$  is a *model* of  $A$
- If  $\mathcal{I}(A) = 0$ , we write  $\mathcal{I} \not\models A$  and say, equivalently, that  $A$  is *false* in  $\mathcal{I}$ ,  $\mathcal{I}$  *falsifies*  $A$ , or  $\mathcal{I}$  is *not a model* of  $A$
- $A$  is *satisfiable* if it is true in *some* interpretation, and is *unsatisfiable* otherwise
- $A$  is *valid*, or a *tautology*, if it is true in *every* interpretation, and is *invalid*, or *falsifiable*, otherwise
- $A$  and  $B$  are *equivalent*, written  $A \equiv B$ , if they have exactly the same models

# Satisfiability, validity, equivalence

Let  $A$  and  $B$  be two formulas and  $\mathcal{I}$  an interpretation

- If  $\mathcal{I}(A) = 1$ , we write  $\mathcal{I} \models A$  and say, equivalently, that  $A$  is *true* in  $\mathcal{I}$ ,  $\mathcal{I}$  *satisfies*  $A$ , or  $\mathcal{I}$  is a *model* of  $A$
- If  $\mathcal{I}(A) = 0$ , we write  $\mathcal{I} \not\models A$  and say, equivalently, that  $A$  is *false* in  $\mathcal{I}$ ,  $\mathcal{I}$  *falsifies*  $A$ , or  $\mathcal{I}$  is *not a model* of  $A$
- $A$  is *satisfiable* if it is true in *some* interpretation, and is *unsatisfiable* otherwise
- $A$  is *valid*, or a *tautology*, if it is true in *every* interpretation, and is *invalid*, or *falsifiable*, otherwise
- $A$  and  $B$  are *equivalent*, written  $A \equiv B$ , if they have exactly the same models



## Satisfiability, validity, equivalence

Let  $A$  and  $B$  be two formulas and  $\mathcal{I}$  an interpretation

- If  $\mathcal{I}(A) = 1$ , we write  $\mathcal{I} \models A$  and say, equivalently, that  $A$  is *true* in  $\mathcal{I}$ ,  $\mathcal{I}$  *satisfies*  $A$ , or  $\mathcal{I}$  is a *model* of  $A$
- If  $\mathcal{I}(A) = 0$ , we write  $\mathcal{I} \not\models A$  and say, equivalently, that  $A$  is *false* in  $\mathcal{I}$ ,  $\mathcal{I}$  *falsifies*  $A$ , or  $\mathcal{I}$  is *not a model* of  $A$
- $A$  is *satisfiable* if it is true in *some* interpretation, and is *unsatisfiable* otherwise
- $A$  is *valid*, or a *tautology*, if it is true in *every* interpretation, and is *invalid*, or *falsifiable*, otherwise
- $A$  and  $B$  are *equivalent*, written  $A \equiv B$ , if they have exactly the same models

# Satisfiability, validity, equivalence

Let  $A$  and  $B$  be two formulas and  $\mathcal{I}$  an interpretation

- If  $\mathcal{I}(A) = 1$ , we write  $\mathcal{I} \models A$  and say, equivalently, that  $A$  is *true* in  $\mathcal{I}$ ,  $\mathcal{I}$  *satisfies*  $A$ , or  $\mathcal{I}$  is a *model* of  $A$
- If  $\mathcal{I}(A) = 0$ , we write  $\mathcal{I} \not\models A$  and say, equivalently, that  $A$  is *false* in  $\mathcal{I}$ ,  $\mathcal{I}$  *falsifies*  $A$ , or  $\mathcal{I}$  is *not a model* of  $A$
- $A$  is *satisfiable* if it is true in **some** interpretation, and is *unsatisfiable* otherwise
- $A$  is *valid*, or a *tautology*, if it is true in every interpretation, and is *invalid*, or *falsifiable*, otherwise
- $A$  and  $B$  are *equivalent*, written  $A \equiv B$ , if they have exactly the same models

## Satisfiability, validity, equivalence

Let  $A$  and  $B$  be two formulas and  $\mathcal{I}$  an interpretation

- If  $\mathcal{I}(A) = 1$ , we write  $\mathcal{I} \models A$  and say, equivalently, that  $A$  is *true* in  $\mathcal{I}$ ,  $\mathcal{I}$  *satisfies*  $A$ , or  $\mathcal{I}$  is a *model* of  $A$
- If  $\mathcal{I}(A) = 0$ , we write  $\mathcal{I} \not\models A$  and say, equivalently, that  $A$  is *false* in  $\mathcal{I}$ ,  $\mathcal{I}$  *falsifies*  $A$ , or  $\mathcal{I}$  is *not a model* of  $A$
- $A$  is *satisfiable* if it is true in *some* interpretation, and is *unsatisfiable* otherwise
- $A$  is *valid*, or a *tautology*, if it is true in *every* interpretation, and is *invalid*, or *falsifiable*, otherwise
- $A$  and  $B$  are *equivalent*, written  $A \equiv B$ , if they have exactly the same models

# Satisfiability, validity, equivalence

Let  $A$  and  $B$  be two formulas and  $\mathcal{I}$  an interpretation

- If  $\mathcal{I}(A) = 1$ , we write  $\mathcal{I} \models A$  and say, equivalently, that  $A$  is *true* in  $\mathcal{I}$ ,  $\mathcal{I}$  *satisfies*  $A$ , or  $\mathcal{I}$  is a *model* of  $A$
- If  $\mathcal{I}(A) = 0$ , we write  $\mathcal{I} \not\models A$  and say, equivalently, that  $A$  is *false* in  $\mathcal{I}$ ,  $\mathcal{I}$  *falsifies*  $A$ , or  $\mathcal{I}$  is *not a model* of  $A$
- $A$  is *satisfiable* if it is true in *some* interpretation, and is *unsatisfiable* otherwise
- $A$  is *valid*, or a *tautology*, if it is true in *every* interpretation, and is *invalid*, or *falsifiable*, otherwise
- $A$  and  $B$  are *equivalent*, written  $A \equiv B$ , if they have exactly the same models

## Satisfiability, validity, equivalence

Let  $A$  and  $B$  be two formulas and  $\mathcal{I}$  an interpretation

- If  $\mathcal{I}(A) = 1$ , we write  $\mathcal{I} \models A$  and say, equivalently, that  $A$  is *true* in  $\mathcal{I}$ ,  $\mathcal{I}$  *satisfies*  $A$ , or  $\mathcal{I}$  is a *model* of  $A$
- If  $\mathcal{I}(A) = 0$ , we write  $\mathcal{I} \not\models A$  and say, equivalently, that  $A$  is *false* in  $\mathcal{I}$ ,  $\mathcal{I}$  *falsifies*  $A$ , or  $\mathcal{I}$  is *not a model* of  $A$
- $A$  is *satisfiable* if it is true in *some* interpretation, and is *unsatisfiable* otherwise
- $A$  is *valid*, or a *tautology*, if it is true in *every* interpretation, and is *invalid*, or *falsifiable*, otherwise
- $A$  and  $B$  are *equivalent*, written  $A \equiv B$ , if they have exactly the same models

# Examples

$A, B$  formulas       $p$  propositional variable

- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all satisfiable
- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all falsifiable
- $A \rightarrow A, A \vee \neg A, A \rightarrow (B \rightarrow A)$  are all valid

# Examples

$A, B$  formulas       $p$  propositional variable

- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all **satisfiable**
- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all falsifiable
- $A \rightarrow A, A \vee \neg A, A \rightarrow (B \rightarrow A)$  are all valid

# Examples

$A, B$  formulas       $p$  propositional variable

- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all **satisfiable**
- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all **falsifiable**
- $A \rightarrow A, A \vee \neg A, A \rightarrow (B \rightarrow A)$  are all **valid**



# Examples

$A, B$  formulas       $p$  propositional variable

- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all **satisfiable**
- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all **falsifiable**
- $A \rightarrow A, A \vee \neg A, A \rightarrow (B \rightarrow A)$  are all **valid**

# Examples

$A, B$  formulas       $p$  propositional variable

- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all **satisfiable**
- $A \rightarrow B, A \vee \neg B, (A \rightarrow B) \rightarrow A, p$  are all **falsifiable**
- $A \rightarrow A, A \vee \neg A, A \rightarrow (B \rightarrow A)$  are all **valid**

## Note:

- $\top$  is valid and  $\perp$  is unsatisfiable
- Every valid formula is satisfiable
- Every unsatisfiable formula is falsifiable

## Examples: equivalences

For all formulas  $A$  and  $B$ , the following equivalences hold:

$$A \rightarrow \perp \equiv \neg A \quad (1)$$

$$\top \rightarrow A \equiv A \quad (2)$$

$$A \rightarrow B \equiv \neg A \vee B \quad (3)$$

$$\equiv \neg(A \wedge \neg B) \quad (4)$$

$$A \wedge B \equiv \neg(\neg A \vee \neg B) \quad (5)$$

$$A \vee B \equiv \neg A \rightarrow B \quad (6)$$

$$A \rightarrow A \equiv \top \quad (7)$$

$$A \wedge \neg A \equiv \perp \quad (8)$$

## Connections between these notions

For all formulas  $A$  and  $B$ ,

1.  $A$  is **valid** iff  $\neg A$  is **unsatisfiable**
2.  $A$  is **satisfiable** iff  $\neg A$  is **falsifiable**
3.  $A$  is **valid** iff  $A$  is **equivalent** to  $\top$
4.  $A$  and  $B$  are **equivalent** iff  $A \leftrightarrow B$  is **valid**
5.  $A$  and  $B$  are **equivalent** iff  $\neg(A \leftrightarrow B)$  is **unsatisfiable**
6.  $A$  is **satisfiable** iff  $A$  is **not equivalent** to  $\perp$

## Connections between these notions

For all formulas  $A$  and  $B$ ,

1.  $A$  is valid iff  $\neg A$  is unsatisfiable
2.  $A$  is satisfiable iff  $\neg A$  is falsifiable
3.  $A$  is valid iff  $A$  is equivalent to  $\top$
4.  $A$  and  $B$  are equivalent iff  $A \leftrightarrow B$  is valid
5.  $A$  and  $B$  are equivalent iff  $\neg(A \leftrightarrow B)$  is unsatisfiable
6.  $A$  is satisfiable iff  $A$  is not equivalent to  $\perp$

## Connections between these notions

For all formulas  $A$  and  $B$ ,

1.  $A$  is valid iff  $\neg A$  is unsatisfiable
2.  $A$  is satisfiable iff  $\neg A$  is falsifiable
3.  $A$  is valid iff  $A$  is equivalent to  $\top$
4.  $A$  and  $B$  are equivalent iff  $A \leftrightarrow B$  is valid
5.  $A$  and  $B$  are equivalent iff  $\neg(A \leftrightarrow B)$  is unsatisfiable
6.  $A$  is satisfiable iff  $A$  is not equivalent to  $\perp$

## Connections between these notions

For all formulas  $A$  and  $B$ ,

1.  $A$  is **valid** iff  $\neg A$  is **unsatisfiable**
2.  $A$  is **satisfiable** iff  $\neg A$  is **falsifiable**
3.  $A$  is **valid** iff  $A$  is **equivalent** to  $\top$
4.  $A$  and  $B$  are **equivalent** iff  $A \leftrightarrow B$  is **valid**
5.  $A$  and  $B$  are **equivalent** iff  $\neg(A \leftrightarrow B)$  is **unsatisfiable**
6.  $A$  is **satisfiable** iff  $A$  is **not equivalent** to  $\perp$

## Syntactic vs. semantic symbols

For all formulas  $A$  and  $B$ ,

- $A$  is valid iff  $A \equiv \top$
- $(A \leftrightarrow B)$  is valid iff  $(A \leftrightarrow B) \equiv \top$

What is the difference between  $\equiv$  and  $\leftrightarrow$ ?

$\leftrightarrow$  is a connective in the language of propositional logic

$\equiv$  is mathematical notation to express formula equivalence



## Syntactic vs. semantic symbols

For all formulas  $A$  and  $B$ ,

- $A$  is valid iff  $A \equiv \top$
- $(A \leftrightarrow B)$  is valid iff  $(A \leftrightarrow B) \equiv \top$

What is the difference between  $\equiv$  and  $\leftrightarrow$ ?

$\leftrightarrow$  is a connective in the language of propositional logic

$\equiv$  is mathematical notation to express formula equivalence

## Syntactic vs. semantic symbols

For all formulas  $A$  and  $B$ ,

- $A$  is valid iff  $A \equiv \top$
- $(A \leftrightarrow B)$  is valid iff  $(A \leftrightarrow B) \equiv \top$

What is the difference between  $\equiv$  and  $\leftrightarrow$ ?

$\leftrightarrow$  is a **connective** in the language of propositional logic

$\equiv$  is **mathematical notation** to express formula equivalence

## Syntactic vs. semantic symbols

For all formulas  $A$  and  $B$ ,

- $A$  is valid iff  $A \equiv \top$
- $(A \leftrightarrow B)$  is valid iff  $(A \leftrightarrow B) \equiv \top$

What is the difference between  $\equiv$  and  $\leftrightarrow$ ?

$\leftrightarrow$  is a **connective** in the language of propositional logic

$(A \leftrightarrow B)$  is **formula** of the logic

$\equiv$  is **mathematical notation** to express formula equivalence

$(A \equiv B)$  is a shorthand for a **statement** about the interpretations of  $A$  and  $B$

## How to evaluate a formula?

Let's evaluate the formula

$$A = (p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$$

in the interpretation

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula	value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$	1
$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$	0
$p \wedge q \rightarrow r$	1
$p \rightarrow q$	0
$p \wedge q$	0
$p$	1
$q$	0
$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula	value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$	1
$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$	0
$p \wedge q \rightarrow r$	1
$p \rightarrow q$	0
$p \wedge q$	0
$p$	1
$q$	0
$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula	value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$	1
$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$	0
$p \wedge q \rightarrow r$	1
$p \rightarrow q$	0
$p \wedge q$	0
$p$	1
$q$	0
$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula				value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$				1
$p \rightarrow r$				1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$				0
$p \wedge q \rightarrow r$				1
$p \rightarrow q$				0
$p \wedge q$				0
$p$	$p$	$p$	$p$	1
$q$	$q$	$q$	$q$	0
	$r$	$r$	$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$



## Evaluating a formula

formula				value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$				1
			$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$				0
			$p \wedge q \rightarrow r$	1
$p \rightarrow q$				0
			$p \wedge q$	0
$p$	$p$	$p$		1
	$q$	$q$		0
		$r$	$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula				value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$				1
$p \rightarrow r$				1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$				0
$p \wedge q \rightarrow r$				1
$p \rightarrow q$				0
$p \wedge q$				0
$p$	$p$	$p$	$p$	1
$q$	$q$			0
		$r$	$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula	value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$	1
$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$	0
$p \wedge q \rightarrow r$	1
$p \rightarrow q$	0
$p \wedge q$	0
$p$	1
$q$	0
$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula	value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$	1
$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$	0
$p \wedge q \rightarrow r$	1
$p \rightarrow q$	0
$p \wedge q$	0
$p$	1
$q$	0
$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula	value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$	1
$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$	0
$p \wedge q \rightarrow r$	1
$p \rightarrow q$	<b>0</b>
$p \wedge q$	0
$p$	1
$q$	0
$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula			value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$			1
		$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$			0
	$p \wedge q \rightarrow r$		1
$p \rightarrow q$			0
	$p \wedge q$		0
$p$	$p$	$p$	1
	$q$	$q$	0
		$r$	1
		$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

# Evaluating a formula

formula				value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$				1
$p \rightarrow r$				1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$				0
$p \wedge q \rightarrow r$				1
$p \rightarrow q$				0
$p \wedge q$				0
$p$	$p$	$p$		1
	$q$	$q$		0
		$r$	$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

## Evaluating a formula

formula			value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$			1
		$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$			0
	$p \wedge q \rightarrow r$		1
$p \rightarrow q$			0
	$p \wedge q$		0
$p$	$p$	$p$	1
	$q$	$q$	0
		$r$	1
		$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$



## Evaluating a formula

formula	value
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$	<b>1</b>
$p \rightarrow r$	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$	0
$p \wedge q \rightarrow r$	1
$p \rightarrow q$	0
$p \wedge q$	0
$p$	1
$q$	0
$r$	1

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

So the formula is **true** in interpretation  $\mathcal{I}$

## Equivalent replacement

Let  $B[A]$  denote a formula  $B$  with a fixed occurrence of a subformula  $A$

Let  $B[A']$  then denote the formula obtained from  $B$  by replacing that occurrence of  $A$  by  $A'$

## Equivalent replacement

Let  $B[A]$  denote a formula  $B$  with a fixed occurrence of a subformula  $A$

Let  $B[A']$  then denote the formula obtained from  $B$  by replacing that occurrence of  $A$  by  $A'$

### Example

$$B = (p_1 \wedge p_2) \vee (p_1 \wedge p_3) \quad (9)$$

$$A = p_1 \wedge p_3 \quad (10)$$

$$A' = p_1 \vee \neg p_4 \quad (11)$$

$$B[A'] = (p_1 \wedge p_2) \vee (p_1 \vee \neg p_4) \quad (12)$$

## Equivalent replacement

Let  $B[A]$  denote a formula  $B$  with a fixed occurrence of a subformula  $A$

Let  $B[A']$  then denote the formula obtained from  $B$  by replacing that occurrence of  $A$  by  $A'$

### Lemma (Equivalent Replacement)

Let  $\mathcal{I}$  be an interpretation and  $\mathcal{I} \models A_1 \leftrightarrow A_2$ . Then  $\mathcal{I} \models B[A_1] \leftrightarrow B[A_2]$ .

## Equivalent replacement

Let  $B[A]$  denote a formula  $B$  with a fixed occurrence of a subformula  $A$

Let  $B[A']$  then denote the formula obtained from  $B$  by replacing that occurrence of  $A$  by  $A'$

### Lemma (Equivalent Replacement)

Let  $\mathcal{I}$  be an interpretation and  $\mathcal{I} \models A_1 \leftrightarrow A_2$ . Then  $\mathcal{I} \models B[A_1] \leftrightarrow B[A_2]$ .

### Theorem (Equivalent Replacement)

Let  $A_1 \equiv A_2$ . Then  $B[A_1] \equiv B[A_2]$ .

## Equivalent replacement

Let  $B[A]$  denote a formula  $B$  with a fixed occurrence of a subformula  $A$

Let  $B[A']$  then denote the formula obtained from  $B$  by replacing that occurrence of  $A$  by  $A'$

### Lemma (Equivalent Replacement)

Let  $\mathcal{I}$  be an interpretation and  $\mathcal{I} \models A_1 \leftrightarrow A_2$ . Then  $\mathcal{I} \models B[A_1] \leftrightarrow B[A_2]$ .

### Theorem (Equivalent Replacement)

Let  $A_1 \equiv A_2$ . Then  $B[A_1] \equiv B[A_2]$ .

Thanks to compositionality!

# A purely syntactic formula evaluation algorithm

Let  $\mathcal{I}$  be an interpretation

## Note:

- If  $\mathcal{I} \models p$  then  $\mathcal{I} \models p \leftrightarrow \top$
- If  $\mathcal{I} \not\models p$  then  $\mathcal{I} \models p \leftrightarrow \perp$

By the previous lemma, we can replace a subformula by a formula with the same value

Hence, we can replace every atom  $p$  by either  $\top$  or  $\perp$ , depending on the value of  $p$  in  $\mathcal{I}$

# A purely syntactic formula evaluation algorithm

Let  $\mathcal{I}$  be an interpretation

**Note:**

- If  $\mathcal{I} \models p$  then  $\mathcal{I} \models p \leftrightarrow \top$
- If  $\mathcal{I} \not\models p$  then  $\mathcal{I} \models p \leftrightarrow \perp$

By the previous lemma, we can **replace a subformula by a formula with the same value**

Hence, we can replace every atom  $p$  by either  $\top$  or  $\perp$ , depending on the value of  $p$  in  $\mathcal{I}$



# A purely syntactic formula evaluation algorithm

Let  $\mathcal{I}$  be an interpretation

**Note:**

- If  $\mathcal{I} \models p$  then  $\mathcal{I} \models p \leftrightarrow \top$
- If  $\mathcal{I} \not\models p$  then  $\mathcal{I} \models p \leftrightarrow \perp$

By the previous lemma, we can **replace a subformula by a formula with the same value**

Hence, we can replace every atom  $p$  by either  $\top$  or  $\perp$ , depending on the value of  $p$  in  $\mathcal{I}$

# Rewrite rules for evaluating a formula

Consider a formula whose atoms consist only of  $\perp$  and  $\top$

Any such formula, other than  $\perp$  and  $\top$ , can be rewritten to a smaller, equivalent formula

## Examples

- $A \rightarrow \top$  is equivalent to  $\top$
- $A \vee \perp$  is equivalent to  $A$

This simplification process can be formalized as a rewrite rule system

## Rewrite rules for evaluating a formula

Consider a formula whose atoms consist only of  $\perp$  and  $\top$

Any such formula, other than  $\perp$  and  $\top$ , can be **rewritten** to a **smaller, equivalent** formula

### Examples

- $A \rightarrow \top$  is equivalent to  $\top$
- $A \vee \perp$  is equivalent to  $A$

This simplification process can be formalized as a rewrite rule system

# Rewrite rules for evaluating a formula

Consider a formula whose atoms consist only of  $\perp$  and  $\top$

Any such formula, other than  $\perp$  and  $\top$ , can be **rewritten** to a **smaller, equivalent** formula

## Examples

- $A \rightarrow \top$  is equivalent to  $\top$
- $A \vee \perp$  is equivalent to  $A$

This simplification process can be formalized as a rewrite rule system

# Rewrite rules for evaluating a formula

Consider a formula whose atoms consist only of  $\perp$  and  $\top$

Any such formula, other than  $\perp$  and  $\top$ , can be **rewritten** to a **smaller, equivalent** formula

## Examples

- $A \rightarrow \top$  is equivalent to  $\top$
- $A \vee \perp$  is equivalent to  $A$

This simplification process can be formalized as a **rewrite rule system**

## Rewrite system for formula evaluation

$$\begin{aligned} \top \wedge \dots \wedge \top &\Rightarrow \top \\ \perp \wedge A_1 \wedge \dots \wedge A_n &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \top \vee A_1 \vee \dots \vee A_n &\Rightarrow \top \\ \perp \vee \dots \vee \perp &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \neg \top &\Rightarrow \perp \\ \neg \perp &\Rightarrow \top \end{aligned}$$

$$\begin{aligned} A \rightarrow \top &\Rightarrow \top \\ \perp \rightarrow A &\Rightarrow \top \\ \top \rightarrow \perp &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \top &\leftrightarrow \top \Rightarrow \top \\ \top &\leftrightarrow \perp \Rightarrow \perp \\ \perp &\leftrightarrow \top \Rightarrow \perp \\ \perp &\leftrightarrow \perp \Rightarrow \top \end{aligned}$$

$\Rightarrow$  is a *rewrite relation*

## Rewrite system for formula evaluation

$$\begin{aligned} \top \wedge \dots \wedge \top &\Rightarrow \top \\ \perp \wedge A_1 \wedge \dots \wedge A_n &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \top \vee A_1 \vee \dots \vee A_n &\Rightarrow \top \\ \perp \vee \dots \vee \perp &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \neg \top &\Rightarrow \perp \\ \neg \perp &\Rightarrow \top \end{aligned}$$

$$\begin{aligned} A \rightarrow \top &\Rightarrow \top \\ \perp \rightarrow A &\Rightarrow \top \\ \top \rightarrow \perp &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \top &\leftrightarrow \top \Rightarrow \top \\ \top &\leftrightarrow \perp \Rightarrow \perp \\ \perp &\leftrightarrow \top \Rightarrow \perp \\ \perp &\leftrightarrow \perp \Rightarrow \top \end{aligned}$$

$\Rightarrow$  is a *rewrite relation*

## Rewrite system for formula evaluation

$$\begin{aligned} \top \wedge \dots \wedge \top &\Rightarrow \top \\ \perp \wedge A_1 \wedge \dots \wedge A_n &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \top \vee A_1 \vee \dots \vee A_n &\Rightarrow \top \\ \perp \vee \dots \vee \perp &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \neg \top &\Rightarrow \perp \\ \neg \perp &\Rightarrow \top \end{aligned}$$

$$\begin{aligned} A \rightarrow \top &\Rightarrow \top \\ \perp \rightarrow A &\Rightarrow \top \\ \top \rightarrow \perp &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \top \leftrightarrow \top &\Rightarrow \top \\ \top \leftrightarrow \perp &\Rightarrow \perp \\ \perp \leftrightarrow \top &\Rightarrow \perp \\ \perp \leftrightarrow \perp &\Rightarrow \top \end{aligned}$$

$\Rightarrow$  is a *rewrite relation*

Writing  $B \Rightarrow B'$  means that  $B$  can be rewritten to  $B'$  in one step using one of the rules above



## Rewrite system for formula evaluation

$$\begin{aligned} \top \wedge \dots \wedge \top &\Rightarrow \top \\ \perp \wedge A_1 \wedge \dots \wedge A_n &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \top \vee A_1 \vee \dots \vee A_n &\Rightarrow \top \\ \perp \vee \dots \vee \perp &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \neg \top &\Rightarrow \perp \\ \neg \perp &\Rightarrow \top \end{aligned}$$

$$\begin{aligned} A \rightarrow \top &\Rightarrow \top \\ \perp \rightarrow A &\Rightarrow \top \\ \top \rightarrow \perp &\Rightarrow \perp \end{aligned}$$

$$\begin{aligned} \top \leftrightarrow \top &\Rightarrow \top \\ \top \leftrightarrow \perp &\Rightarrow \perp \\ \perp \leftrightarrow \top &\Rightarrow \perp \\ \perp \leftrightarrow \perp &\Rightarrow \top \end{aligned}$$

$\Rightarrow$  is a *rewrite relation*

A formula  $A$  is in *normal form* (wrt  $\Rightarrow$ ) if it cannot be rewritten by any of the rules above

## A syntactic evaluation algorithm

*evaluate* evaluates any formula  $G$  in any interpretation  $\mathcal{I}$  using the previous rewrite system

```
procedure evaluate( $G, \mathcal{I}$ )  
input: formula  $G$ , interpretation  $\mathcal{I}$   
output: the boolean value  $\mathcal{I}(G)$   
begin  
  forall atoms  $p$  occurring in  $G$   
    if  $\mathcal{I} \models p$   
      then replace all occurrences of  $p$  in  $G$  by  $\top$   
    else replace all occurrences of  $p$  in  $G$  by  $\perp$   
  rewrite  $G$  into a normal form using the rewrite rules  
  if  $G = \top$  then return 1 else return 0  
end
```

## A syntactic evaluation algorithm

*evaluate* evaluates any formula  $G$  in any interpretation  $\mathcal{I}$  using the previous rewrite system

```
procedure evaluate( $G, \mathcal{I}$ )  
input: formula  $G$ , interpretation  $\mathcal{I}$   
output: the boolean value  $\mathcal{I}(G)$   
begin  
  forall atoms  $p$  occurring in  $G$   
    if  $\mathcal{I} \models p$   
      then replace all occurrences of  $p$  in  $G$  by  $\top$   
      else replace all occurrences of  $p$  in  $G$  by  $\perp$   
      rewrite  $G$  into a normal form using the rewrite rules  
      if  $G = \top$  then return 1 else return 0  
end
```

## A syntactic evaluation algorithm

*evaluate* evaluates any formula  $G$  in any interpretation  $\mathcal{I}$  using the previous rewrite system

```
procedure evaluate( $G, \mathcal{I}$ )  
input: formula  $G$ , interpretation  $\mathcal{I}$   
output: the boolean value  $\mathcal{I}(G)$   
begin  
  forall atoms  $p$  occurring in  $G$   
    if  $\mathcal{I} \models p$   
      then replace all occurrences of  $p$  in  $G$  by  $\top$   
      else replace all occurrences of  $p$  in  $G$  by  $\perp$   
  rewrite  $G$  into a normal form using the rewrite rules  
  if  $G = \top$  then return 1 else return 0  
end
```

## A syntactic evaluation algorithm

*evaluate* evaluates any formula  $G$  in any interpretation  $\mathcal{I}$  using the previous rewrite system

```
procedure evaluate( $G, \mathcal{I}$ )  
input: formula  $G$ , interpretation  $\mathcal{I}$   
output: the boolean value  $\mathcal{I}(G)$   
begin  
  forall atoms  $p$  occurring in  $G$   
    if  $\mathcal{I} \models p$   
      then replace all occurrences of  $p$  in  $G$  by  $\top$   
    else replace all occurrences of  $p$  in  $G$  by  $\perp$   
  rewrite  $G$  into a normal form using the rewrite rules  
  if  $G = \top$  then return 1 else return 0  
end
```

## Example

Let us evaluate the formula

$$G = (p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$$

in the interpretation

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

Its value is equal to the value of

$$(T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T)$$

## Example

Let us evaluate the formula

$$G = (p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$$

in the interpretation

$$\mathcal{I} = \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

Its value is equal to the value of

$$(T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T)$$

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (\perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge T \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow T \Rightarrow \\ & T \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $T \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow T \Rightarrow T$

Outside-in, right-to-left:

$$(T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T)$$

The result will always be the same independently of the order of rewriting!



# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (\perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge T \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow T \Rightarrow \\ & T \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $T \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow T \Rightarrow T$

Outside-in, right-to-left:

$$(T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (\perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge T \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow T \Rightarrow \\ & T \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $T \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow T \Rightarrow T$

Outside-in, right-to-left:

$$(T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \quad \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \quad \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \quad \perp \rightarrow \top \Rightarrow \\ & \quad \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!



# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \quad \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$(\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top)$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp \wedge (T \wedge \perp \rightarrow T) &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp \wedge (\perp \rightarrow T) &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp \wedge T &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp &\rightarrow T \Rightarrow \\ T & \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $T \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow T \Rightarrow T$

Outside-in, right-to-left:

$$\begin{aligned} (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) &\rightarrow (T \rightarrow T) \Rightarrow \\ (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) &\rightarrow T \Rightarrow \\ T & \end{aligned}$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp \wedge (T \wedge \perp \rightarrow T) &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp \wedge (\perp \rightarrow T) &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp \wedge T &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp &\rightarrow (T \rightarrow T) \Rightarrow \\ \perp &\rightarrow T \Rightarrow \\ T & \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $T \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow T \Rightarrow T$

Outside-in, right-to-left:

$$\begin{aligned} (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) &\rightarrow (T \rightarrow T) \Rightarrow \\ (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) &\rightarrow T \Rightarrow \\ T & \end{aligned}$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (\perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge T \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow T \Rightarrow \\ & T \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $T \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow T \Rightarrow T$

Outside-in, right-to-left:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow T \Rightarrow \\ & T \end{aligned}$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (\perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge T \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow T \Rightarrow \\ & T \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $T \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow T \Rightarrow T$

Outside-in, right-to-left:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow T \Rightarrow \\ & T \end{aligned}$$

The result will always be the same independently of the order of rewriting!



# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge (\perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \wedge T \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow (T \rightarrow T) \Rightarrow \\ & \perp \rightarrow T \Rightarrow \\ & T \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $T \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow T \Rightarrow T$

Outside-in, right-to-left:

$$\begin{aligned} & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow (T \rightarrow T) \Rightarrow \\ & (T \rightarrow \perp) \wedge (T \wedge \perp \rightarrow T) \rightarrow T \Rightarrow \\ & \quad \quad \quad T \end{aligned}$$

The result will always be the same independently of the order of rewriting!

# Apply rewrite rules

Inside-out, left-to-right:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge (\perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \wedge \top \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow (\top \rightarrow \top) \Rightarrow \\ & \perp \rightarrow \top \Rightarrow \\ & \quad \top \end{aligned}$$

1.  $A \wedge \perp \Rightarrow \perp$
2.  $\top \rightarrow \perp \Rightarrow \perp$
3.  $A \rightarrow \top \Rightarrow \top$

Outside-in, right-to-left:

$$\begin{aligned} & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow (\top \rightarrow \top) \Rightarrow \\ & (\top \rightarrow \perp) \wedge (\top \wedge \perp \rightarrow \top) \rightarrow \top \Rightarrow \\ & \quad \top \end{aligned}$$

The result will always be **the same** independently of the **order of rewriting!**